

TARTU ÜLIKOOL

Matemaatika ja informaatika teaduskond

Arvutiteaduse instituut

Madis Pärna

**Töökeskonna Sage kasutamine üldhariduskoolide
matemaatikas**

Bakalaureusetöö

(6 eap)

Juhendaja: prof. Eero Vainikko

Autor „juuni 2011

Juhendaja „juuni 2011

Professor „juuni 2011

Tartu 2011

Sisukord

Sissejuhatus.....	3
1. Sage'i tutvustus.....	4
2. Sage'i tööleht.....	6
3. Sage'i kasutamise näited	9
3.1 Võrrandi lahendamine.....	9
3.2 Funktsiooni uurimine. Muutumis- ja määramispiirkonnad.....	10
3.3 Funktsiooni positiivsus- ja negatiivsuspiirkonnad.....	12
3.4 Paaris- ja paaritu funktsioon.....	13
3.5 Funktsioonide sõltuvuse uurimine.....	15
3.6 Arvu juurimine.....	17
3.7 Integraali leidmine.....	18
3.8 Kõvertrapetsi pindala.....	19
3.9 Kahe joone vaheline pindala.....	21
4. Sage'i server.....	22
Kokkuvõte.....	24
Summary.....	25
Kasutatud kirjandus.....	26

Sissejuhatus

Uurides võimalusi, kuidas leida matemaatiliste probleemide lahendamiseks tarkvara, võib leida mitmeid programme, nagu Maple, Mathematica jne., kus on võimalik väga võimsaid arvutusi teha, analüüsida jooniseid, jms. Need kõik on kahtlemata suurepärased töövahendid, kuid ka suhteliselt kallid. Teades üldhariduskoolide rahalisi võimalusi, ei saa koolid endale lubada tasulisi programme, mis piirab matemaatika õppimist arvutitel. Sellest tulenevalt otsitakse võimalusi, kuidas saaks seda tühimikku täita mõne vabavaralise programmiga.

Internetis on pakkuda mitmeid vabavaralisi matemaatilisi programme, nagu Maxima, SciLab, R jne., mis on mõeldud teatud ülesannete lahendamiseks. Need kõik on väga head, kuid üksikuna ei saa tasuliste programmidega võistelda, mis viib soovini liita kõik head vabavaralised programmid ühe keskse kasutajaliidese alla. Üheks selliseks näiteks on Sage.

Sage ühendab endas enamiku vabavaralisi tarkvarasid ja lisab omalt poolt veel uusi algoritme ning võimalusi. Sage on samuti vabavaraline ja arendajad on seadnud endale ülesandeks muuta uute matemaatika algoritmide rakendamise efektiivsemaks ning tagada käsureal kui ka veebiliidesel töötamine.

Edaspidi soovin avastada Sage'i ja selle võimalusi ning kas seda saaks kasutada koolis. Mõnedes kõrgkoolides on Sage kasutusele võetud, kuid vähe on see levinud üldhariduskoolides, Eestis minu teada mitte kusagil. Sage sisaldab paljusid matemaatika elemente, mida õpetatakse keskkoolis.

Järgnevas peatükis kirjeldan lähemalt Sage'i ülesehitust, omapärasid ja kasutuslihtsustusi. Sage keskkonnas on õpetus ja abi kergesti leitav ning ka koodinäited olemas. Edasi näitan Sage'i võimalusi keskkooli matemaatika ülesannete lahendamisel ja jooniste uurimisel. Sage'is on väga palju võimalusi ning samuti on keskkooli matemaatika väga mahukas - seega esitan paar Sage'i näidet teatud matemaatika-ülesannetele. Kolmandaks koostan töölehe järgnevates peatükkides toodud näidete põhjal. Lõpuks tutvustan ka, kuidas sai Sage keskkond üles seatud C.R. Jakobsoni nim. Gümnaasiumi serveris aadressil <https://193.40.120.134:8050/>.

Järgnevalt tutvustatakse lühidalt Sage'i ja selle võimalusi.

1. Sage'i tutvustus

Sage'i veebilehelt [1] võib lugeda järgnevat:

Sage on tasuta avatud lähtekoodiga matemaatika tarkvara ja GPL litsentsiga. Sage ühendab paljusid olemasolevaid avatud lähtekoodiga pakette Pythonil baseeruvaks kasutajaliideseks.

Missioon: luua korralik tasuta avatud lähtekoodiga alternatiiv programmidele Magma, Maple, Mathematica ja Matlab.

Sage'i esimene versioon avaldati 24. veebruaril 2005. Sellest alates on sajad programmeerijad täiendanud Sage'i ja loonud uusi võimalusi.

Kui mitmed võimsad programmid on tasulised ning Eesti oludes seetõttu kasutajatele kättesaamatud, siis Sage on täiesti tasuta kättesaadav. Tasulisi programme arendavad teatud firmad, mis võrreldes Sage'i avatud lähtekoodiga ei ole nii kasutajasõbralik. Sage'i arendamisse saavad kaasa aidata kõik soovijad, kes tunnevad programmeerimiskeelt Python, milles on programmeeritud Sage. Seega ka Sage'i kasutamisel on hea tunda Python programmeerimiskeelt, kuid piisab ka mõne muu programmeerimiskeele oskusest.

Sage sisaldab ligi 100 erinevat avatud lähtekoodiga paketti, mis on ühendatud ühe kasutajaliidesega [1]. Avatud lähtekoodiga pole ainukesena jsMath pakett, mis loob veebis matemaatiliste valemite kuvamise võimaluse.

Sage'is sisalduvad matemaatilised tarkvarapaketid [8]:

Algebra	GAP [16], Maxima [17], Singular [18]
Algebraalne geomeetria	Singular [18]
Aritmeetika	MPIR [19], MPFR [20], NTL [21], mpmath [22]
Aritmeiline geomeetria	PARI/GP [23], NTL [21], mwrank [24]
Elementaarne arvutamine	Maxima [17], SymPy [25], GiNaC [26]
Kombinatorika	Symmetica [27]
Lineaarne algebra	ATLAS [28], BLAS [29], LAPACK [30], NumPy [31], LinBox [32], IML [33], GSL [34]
Graafiteooria	NetworkX [35]
Hulgateooria	GAP [16]
Numbriline arvutamine	GSL [34], SciPy [36], NumPy [31], ATLAS [28]
Numbriteooria	PARI/GP [23], FLINT [37], NTL [21]
Statistika	R [38], SciPy [36]

Tabel 1: tarkvarapaketid

Sage'i arendajate jätkuvaks eesmärgiks on tagada kõigi komponentide probleemideta ja efektiivne koostöö ning täiendada olemasolevaid matemaatilisi algoritme ja rakendusi.

Sage on väga mitmekülgne matemaatiliste arvutuste tegemise keskkond.

Sage'i server töötab Ubuntu Linux baasil ehk siis Windowsi arvuti peal saab seda virtuaalmasinas käima panna. Sage'i käivitatakse terminali kätse, andes käsu `sage`. Sage'i käsureal saab olemasolevaid pakette installeerida ja uuendada. Käsurida on hea kasutada inimestel, kes on harjunud seal tegutsema, kuid vähem kogenul on hõlpsam kasutada graafilise liidesega Sage'i. Peale `notebook()` käsu andmist, saab Sage'i kasutada veebiliidesega. Kui käsule `notebook()` määrata täpsemad veebiaadressi parameetrid, saab keskkonnale ligi ka teistest kohtvõrgu arvutitest [11].

Sage'i veebiliidest saab kasutada, logides kasutajana sisse või vaadata anonüümselt avalikkusele publitseeritud Sage materjale. Sage'i täielikuks kasutamiseks peab sisse logima, misjärel avaneb Sage'i töökeskkond, kus saab luua uusi töölehti (ik. *worksheet*) ning uurida ja katsetada teiste poolt publitseeritud. Sage'i veebiliideses on olemas ka paljude teemade kohta õpetusi ja koodinäiteid, mida saab kopeerida enda loodud töölehele.

Sage'is saab koostada hästi mitmekesiseid töölehti, mis koosnevad lineaarselt järjestatavatest lahtritest, kuhu saab sisestada arvutuskäike, koodi ja teksti, mida Sage töötleb. Lahtrid on alguses üherealised, kuid suurenevad uue rea lisamisel. Lahtrite seis salvestatakse failisüsteemi.

Töölehe lahtri töötlemiseks on vaja vajutada klahvikombinatsiooni `<shift-enter>` või vajutada hiirega lahtri vasakul all servas olevale lingile „*evaluate*“. `<Enter>` klahvi kasutatakse nagu tavalises tekstitöötlusprogrammis uue rea loomiseks. Lahtri sisu töötlemise ajaks kuvatakse vasakul servas rohelist vertikaalset püstkriipsu. Kui anti töötlemiseks töölehe kõige alumine lahter, siis luuakse automaatselt uus lahter viimase lahtri alla. Loomaks uut lahtrit aktiivse lahtri alla peab vajutama klahve `<alt-enter>`, vajutades `<ctrl-enter>`, jagatakse kursori asukohast aktiivne lahter kaheks. Lahtri kustutamiseks peab kogu lahtri sisu kustutama ning tühjas üherealises lahtris veelkord klahvi `<backspace>` vajutama.

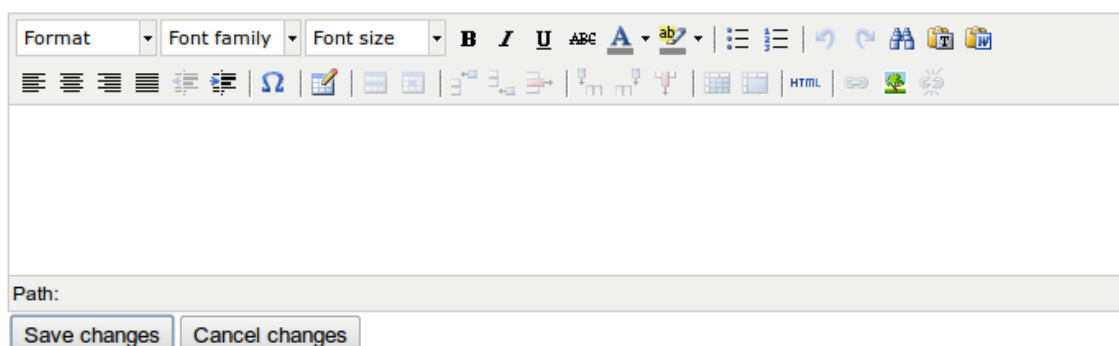
Sage'i töölehel on võimalik funktsioonide kohta abi informatsiooni saada, sisestades funktsiooni järgi sümboli `?` ning vajutades klahvi `<TAB>` või `<shift-enter>`. Tulemuseks kuvatakse funktsiooni täpsem kasutusjuhend. Topeltküsimärkide abil saab täpsemat funktsiooni infot juba lähtekoodi vormis. Funktsiooni abi HTML kujul saamiseks on vaja käsule `help()` anda argumendiks funktsiooni nimi, mis avab uue brauseri akna või saki funktsiooni infoga. Kui funktsiooni nime kuju ei meenu täielikult, siis sisestada funktsiooni esitähed ning vajutada `<TAB>` klahvi, mis kuvab kõik selliselt algavad funktsioonid. [9]

Sage'is saab luua erinevaid graafilisi objekte, nii 2D kui ka 3D graafikuid.

Järgnevas peatükis vaatame, kuidas saab kasutada Sage'i keskkooli matemaatika õppimisel.

2. Sage'i tööleht

Sage'i töölehe loomisel saab esmalt sisestada nime (näiteks läbitava matemaatika osa nimetus), mille järel avaneb ühe lahtriga tööleht. Tavaliselt koostatakse matemaatikas teemat tutvustav tekst koos näidisvalemitega ja lahenduskäikudega, mis on ka Sage'is teostatavad. Samuti saab teksti lisada iga ülesande vahele. Teksti lisamiseks peab kursori viima lahtri ülaserba, kus siis tekib sinine paks horisontaaljoon, ning hoides all klahvi <shift> klikkida seal. Tulemusena ilmub nähtavale tekstisisestuslahter (illustratsioon 1), mille üleval servas on vormingu elemendid.



Illustratsioon 1: tekstisisestuslahter

Sellesse lahtrisse saab lisada pilte ja linke ning korrektsel kujul matemaatika avaldise ja lahendusi. Tihti on vaja lisada teksti ka erisümboleid, näiteks kreeka tähti ja muid matemaatilisi tehtmärke, mis pole leitavad klaviatuurilt. Selle teostamiseks saab Sage'is kasutada LaTeX vormindust.

Kõige lihtsama ruutvõrrandi koostamiseks pole vaja kasutada erilisi LaTeX sümbolmärke. Valem on vaja lihtsalt kirjutada dollarmärkide vahele. Kui valemi ümber panna topelt dollarmärgid, siis joondatakse valem tekstilahtri keskele eraldiseisva valemina. Sisestades valemi kujul $x^2+2x+12=0$ saab tulemuseks:

$$x^2 + 2x + 12 = 0$$

Illustratsioon 2: ruutvõrrand

Toon ühe näite. Oletame, et meil on vaja meeldetuletuseks näidata, mis on tulemuseks, kui reaalarvul a on reaalarvulise astendajaga aste [4, lk 87]. Teksti lahtrisse minev kood on järgmisel kujul:

$a^0 = 1$ \text{, kui} $a \neq 0$

$a^1 = a$

$a^n = \underbrace{a * a * \dots * a}_n$ \text{n tegurit} \text{, kui } n \in \{2; 3; 4; \dots\}

$a^{-k} = \frac{1}{a^k}$ \text{, kui } a \neq 0 \text{ ja } k \in \mathbb{Z} \text{ või } a > 0 \text{ ja } k \in \mathbb{Q}

$a^{m/n} = \begin{cases} \sqrt[n]{a^m} & \text{, kui } a > 0, m \in \mathbb{Z} \\ 0 & \text{, kui } a = 0, m \in \mathbb{Z}^+ \text{ ja } n \in \mathbb{Z}^+ \end{cases}$

Tulemuseks on:

$$a^0 = 1, \text{ kui } a \neq 0$$

$$a^1 = a$$

$$a^n = \underbrace{a * a * \dots * a}_n, \text{ kui } n \in \{2; 3; 4; \dots\}$$

$$a^{-k} = \frac{1}{a^k}, \text{ kui } a \neq 0 \text{ ja } k \in \mathbb{Z} \text{ või } a > 0 \text{ ja } k \in \mathbb{Q}$$

$$a^{m/n} = \begin{cases} \sqrt[n]{a^m}, & \text{kui } a > 0, m \in \mathbb{Z} \text{ ja } n \in \mathbb{Z}^+ \\ 0, & \text{kui } a = 0, m \in \mathbb{Z}^+ \text{ ja } n \in \mathbb{Z}^+ \end{cases}$$

Valemite koostamisel on kasutatud LaTeX käske. Üheks oluliseks käsuks on `\text{}`, millega saab valemite sisse kirjutata teksti, ning käsk `;` loomaks tühikuid valemite elementide vahele. Järgmisena on matemaatika valemites vajalik klamber näiteks mitme rea või mitme elemendi kokkusidumiseks, milleks on käsud `\underbrace` alumiseks klambriks, `\overbrace` ülemiseks klambriks ning `\lbrace` ja `\rbrace` vastavalt vasakuks ja paremaks klambriks. Mitmerealise valikvariantide jaoks on vaja kasutada käske `\begin{cases}` ja `\end{cases}`. Mittevõrdumist tähistab käsk `\neq`, kuuluvust `\in` ning ruutjuure leidmiseks `\sqrt{}`, kus esimeseks argumendiks on juurimisarv ja teiseks juurealune.

Võib täheldada, et kõik LaTeX käsud algavad vasakkaldkriipsuga. LaTeX'i matemaatilisi tehtemärke ja sümboleid on väga palju. Lühike ülevaade levinumatest matemaatika käskudest võib leida näiteks aadressil <http://ftp.ams.org/ams/.../amsmath/short-math-guide.pdf> [10].

Edaspidi vaatame arvutuskäikude sisestamist töölehe lahtritesse. Kui tööleht on valmis, siis tuleb vajutada nuppu „*save and quit*“, mille tulemusel tööleht salvestatakse ja minnakse tagasi töölehtede nimekirja vaatesse. Võimaldamaks teistel näha ja soovi korral ka lahendada Sage'is koostatud ülesandeid, peab valima kas „*publish*“ või „*share*“. *Publish* puhul muudetakse valitud tööleht nähtavaks kõigile kasutajatele, kes

Sage'i avalehele satuvad ja klikivad lingil „*Browse published Sage worksheets*“, avaneb leht, kus on kõik vastavas Sage'i serveris publitseeritud töölehed. Neid töölehti saab lihtsalt lugeda ja uurida, kuid muuta ja käivitada koode ei saa. Töölehe avamiseks peab Sage'i sisse logima ning eelnevalt publitseeritud loendist alla laadima ja personaalsesse töölehtede nimekirja üles laadima. Selliselt saab endale katsetamiseks alla laadida töölehti paljudest Sage'i avalikest serveritest üle maailma.

Teiseks võimaluseks on töölehe jagamine (ik. *share*) valitud kasutajatega. Sel juhul jääb ära alla-ja üleslaadimine, sest kasutaja logib Sage'i sisse ja loob väljajagatud töölehest koopia.

3. Sage'i kasutamise näited

Koolimatemaatika õppematerjalide loomine on keeruline ülesanne. Matemaatika käsitus on nõudlik ka tüpograafiliselt – vaja on kasutada matemaatilisi valemehid, kus on näiteks integraali märke, kreeka tähestiku sümboleid jne. Väga oluline roll heade materjalide loomisel on ka illustratsioonidel. Toon edaspidi mõne näite konkreetsete matemaatika ülesannete baasil, mis on mõeldud tutvustama võimalusi, mida Sage pakub. Õppemetoodiliselt läbimõeldud korrektsete interaktiivsete töölehtede koostamine on võimalik tulevikuülesanne.

Olulisim, mida keskkooli matemaatikas läbitakse, on võrrandite, võrratuste ja võrrandi- ning võrratussüsteemide lahendamine. Võrrandeid ja võrratusi on erinevat tüüpi: algebralised, trigonomeetriselised, eksponent, logaritm. Kõigi nende lahendamine on Sage'is võimalik määramata lahendama hakkamisel selle tüüpi.

Sage sisaldab kõiki trigonomeetriselisi funktsioone, nagu siinus, koosinus, tangens jne. Lisaks veel eksponent ja logaritm, mida tähistatakse vastavalt \exp ja \log .

3.1 Võrrandi lahendamine

Järgnevalt esitan ühe näite võrrandi lahendamisest, mis on võetud gümnaasiumi matemaatika lõpueksamiks valmistumise ülesannetest [6, lk 11].

Algebralised võrrandid

Ülesanne 73.2

```
x = var('x')
fx = 6/(x^2 - 9) + 2/(x^2 + 4*x) == 7/(x^2 + x - 12)
vastus1 = solve(fx, x)
fx; vastus1
```

$$\frac{6}{x^2-9} + \frac{2}{x^2+4x} = \frac{7}{x^2+x-12}$$
$$[x = (-6)]$$

Ülesanne 70.2

```
fa = 5/(3-x) - 1 == 1/(x-3) - 2/(2*x^2-18)
vastus2 = solve(fa, x)
fa; vastus2
```

$$-\frac{5}{x-3} - 1 = \frac{1}{x-3} - \frac{1}{x^2-9}$$
$$[x = (-2), x = (-4)]$$

Illustratsioon 3: algebralised võrrandid

Illustratsioonis 3 on kujutatud algebraliste võrrandite lahendamist. Kasutamaks muutujat x valemis ja lahendites, peab selle eelnevalt defineerima `x=var('x')`. Edasi mõlema ülesande puhul on muutujatele $fb1$ ja $fb2$ antud väärtuseks lahendamist vajavad võrrandid. Funktsioonide kirjapanekul peab jälgima, et sulud saaksid õigesti kohtadesse ning et muutujale väärtuse andmisel kasutatakse ühekordset võrdusmärki, kuid võrrandi kahe poole võrdsuse defineerimiseks on vaja kasutada kahekordset võrdusmärki.

Kui vajalikud muutujad on defineeritud, siis kasutame käsku `solve()`, mille argumentideks on funktsioon ja muutuja. Põhimõtteliselt pole vahet, kas lahendada ühte võrrandit või võrrandisüsteemi. Viimase puhul on kasulik defineerida mitu võrrandit ning käsule `solve()` esimese argumentina anda kantsulgudes komadega eraldatud võrrandite loendi ning teisteks argumentideks muutujad, mis esinevad võrrandites ning mille suhtes tuleb lahendada.

Võrrandite ja nende lahendite kuvamiseks on sisestatud uuele reale semikoolonitega eraldatud eeldefineeritud muutujad. Põhimõtteliselt võiks lahtri koodi ühele reale viia ning semikoolonitega eraldada, kuid siis poleks sooritatavad tehted selgesti eristatavad.

Järgmisena vaatame illustratsioonis 4 võrratuste lahendamist võrratussüsteemina. [6, lk 18]

Ülesanne 192.2

```
fb1 = 3*x^2 - 45*x < 0
fb2 = abs(4 - 2*x) >= 24
vastus3 = solve([fb1, fb2], x)
fb1; fb2; vastus3
```

$$\begin{aligned} 3x^2 - 45x < 0 \\ |-2x + 4| \geq 24 \\ [[14 < x, x < 15], [x = 14]] \end{aligned}$$

Illustratsioon 4: võrratussüsteemi lahendamine

Samuti nagu võrrandite puhul on defineeritud võrratused muutujatele $fb1$ ja $fb2$. Sage'is pole funktsioonide tähistamisel vahet. Teises võrratuses on kasutusel funktsiooni `abs()`, mis leiab etteantud argumenti absoluutväärtuse.

3.2 Funktsiooni uurimine. Muutumis- ja määramispiirkonnad.

Ühe suure osa keskkooli matemaatikast moodustab funktsioonide ja nende graafikute uurimine. Keskkoolis vaadeldakse funktsiooni kujul $y=f(x)$, $x \in X$ ja $y \in Y$ ja f on

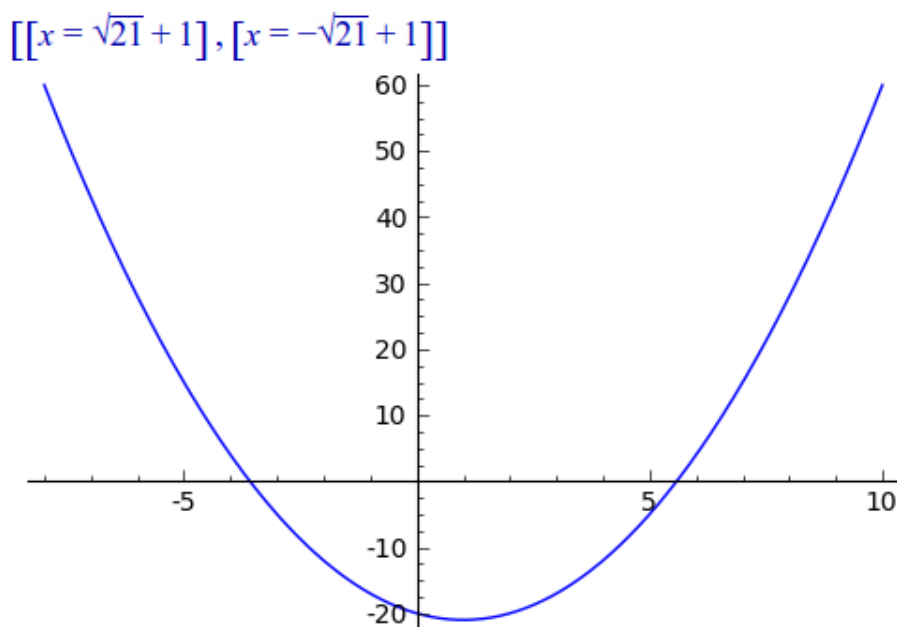
eeskiri, mille kohaselt argumendi igale väärtusele seatakse vastavusse funktsiooni väärtus.

Esmalt tuleb funktsiooni uurimisel välja selgitada funktsiooni määramis- ja muutumispiirkond.

Õpiku definitsioon ütleb:

Funktsiooni määramispiirkond X on sõltumatu muutuja ehk argumendi x väärtuste hulk. Funktsiooni muutumispiirkond Y on sõltuva muutuja y väärtuste ehk funktsiooni väärtuste hulk. [2, lk 75, 76]

```
fd1 = x^2-2*x-20
null = solve([fd1, fd1==0], x)
joonis = plot(fd1, -8, 10)
null; joonis
```



Illustratsioon 5: nullkohad, positiivsus- ja negatiivsuspierikond

Funktsiooni nullkohtade leidmiseks on vaja käsu `solve()` esimeseks argumendiks anda loend, kus üheks elemendiks on funktsioon ise ja teiseks funktsiooni väärtuse võrdumine nulliga. Positiivsus- ja negatiivsuspierikondade väljaselgitamiseks tuleb teha joonis, mida Sage'is sooritab käsk `plot()`, kus esimeseks argumendiks on funktsioon ning teiseks määratud x -vahemik, kus joonis kuvatakse. Kui x -vahemik pole ette antud, näidatakse funktsiooni graafikut vahemikus $[-1, 1]$. Mitme funktsiooni ühele koordinaatteljestikule joonistamiseks on vaja käsule `plot()` anda esimeseks argumendiks funktsioonide loend.

3.3 Funktsiooni positiivsus- ja negatiivsuspiirkonnad

Funktsiooni positiivsuspiirkonnad moodustavad argumendi need väärtused, mille korral funktsiooni väärtus on positiivne. Funktsiooni positiivsuspiirkonna leidmiseks tuleb määrata punktid x , kus $f(x) > 0$. [4, lk 59]

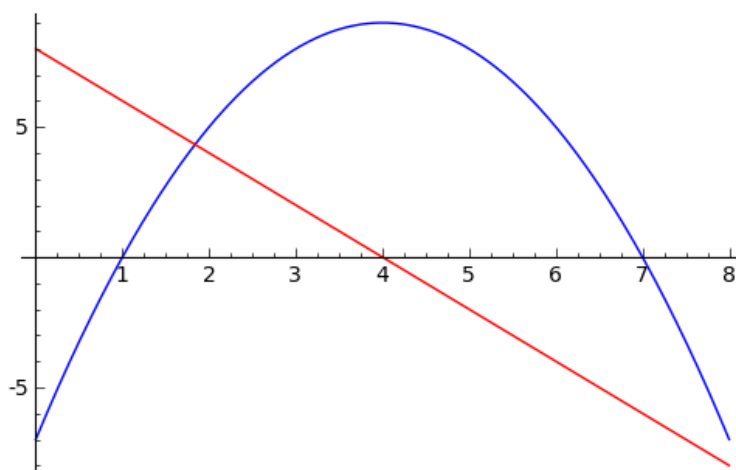
Seega negatiivsuspiirkond on punkti x väärtuste hulk, kus $f(x) < 0$. Kuna nullkohad on käes ja joonis tehtud, saab määrata positiivsus- ja negatiivsuspiirkonnad.

Funktsiooni uurimisel vaatame järgmisena, kuidas leida kasvamis- ja kahanemisvahemikke.

Funktsiooni kasvamine ja kahanemine. Funktsiooni ekstreemumkohad

```
ff1(x) = -(x-1)*(x-7)
ff1_joonis = plot(ff1, 0,8)
null = solve([ff1, ff1==0],x)
dff1 = diff(ff1,x)
dff1_joonis = plot(dff1, 0,8, color='red')
dff1x = solve(dff1,x)
ff1; ff1_joonis + dff1_joonis; null; dff1; dff1x; ff1(4)
```

$$x \mapsto -(x-7)(x-1)$$



$$[[x = 1], [x = 7]]$$

$$x \mapsto -2x + 8$$

$$[x = 4]$$

$$9$$

Illustratsioon 6: funktsiooni kasvamine ja kahanemine

Funktsiooni $y=f(x)$ nimetatakse kasvavaks vahemikus (a,b) , kui selles vahemikus argumendi väärtuste suurenedes ka funktsiooni vastavad väärtused suurenevad: kui $x_1 < x_2$, siis ka $f(x_1) < f(x_2)$. Funktsiooni $y=f(x)$ nimetatakse kahanevaks vahemikus (a,b) , kui selles vahemikus argumendi väärtuste suurenedes funktsiooni

vastavad väärtused vähenevad: kui $x_1 < x_2$, siis $f(x_1) > f(x_2)$. [4, lk 61]

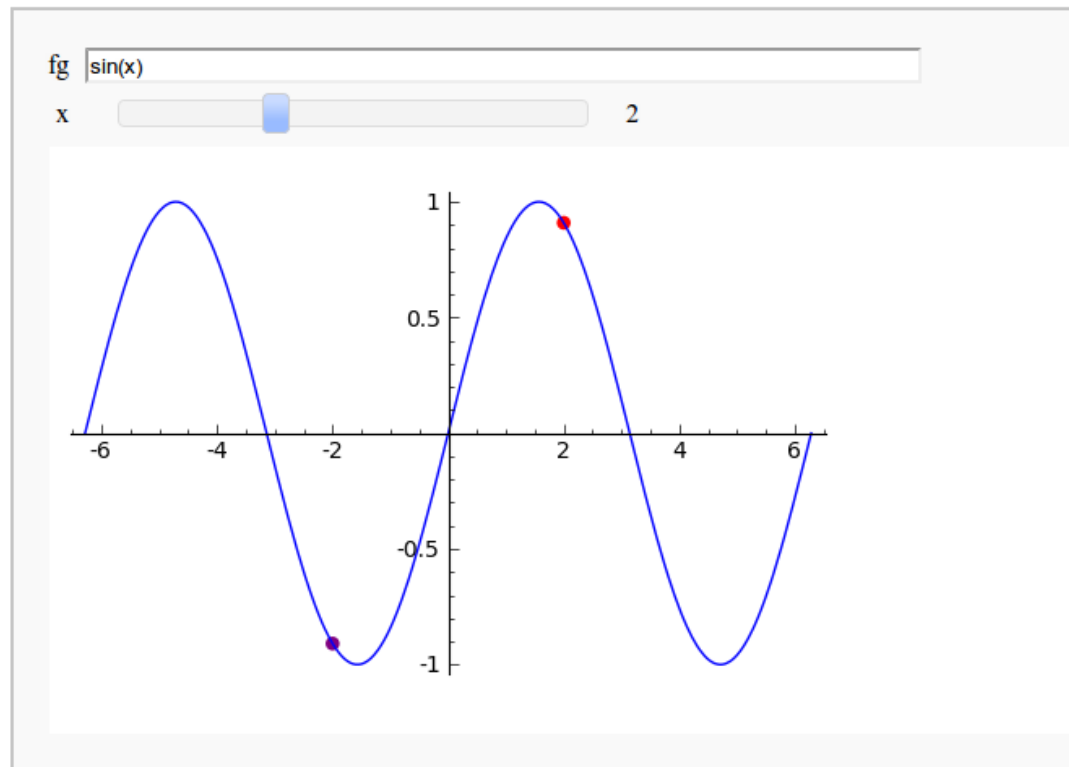
Illustratsioonil 6 on toodud näiteks funktsioon $ff1(x)$ ning on leitud nullkohad, et saaks joonistada graafikut õiges muutujate vahemikus. Nullkohtadeks on $x=1$ ja $x=7$. Ekstreemumkohtade leidmiseks peame funktsioonist leidma esimese tuletise, mida Sage'is saab teha käsuga `diff()` või `derivative()`. Käsu `diff()` esimeseks argumendiks on funktsioon ja teiseks, mis muutuja suhtes tuletis leitakse. Funktsiooni esimene tuletis on $-2x+8$ ning lahendades esimest tuletist, saame, et ekstreemumkoht on $x=4$. Ekstreemumi saamiseks peame panema ekstreemumkoha väärtuse algsesse funktsiooni ning saame selleks üheksa. Saamaks paremat ülevaadet funktsioonide seostest saab Sage'is mitu graafikut ühele koordinaatteljestikule kanda. Antud juhul on kujutatud sinise joonena algfunktsioon ja punase joonena selle esimene tuletis. Graafiku joone värvi muutmiseks peab käsule `plot()` üheks argumendiks andma `color` või `rgbcolor`, mille väärtuseks võivad olla vastavalt kas ingliskeelne värvinimetus või html värvikood.

Kasvamis- ja kahanemisvahemiku saamiseks on meil juba välja arvatud ekstreemumkoht ja samuti tehtud joonis. Kasvamisvahemik on $x < 4$ ja kahanemisvahemik $x > 4$. Funktsiooni maksimum- ja miinimumkohtade määramiseks on samuti leitud juba ekstreemumkoht. Kui funktsiooni väärtus ekstreemumkohal on vähim funktsiooni väärtus, siis nimetatakse ekstreemumkohta miinimumkohaks, vastasel korral maksimumkohaks. Antud funktsiooni $ff1$ puhul on tegemist maksimumkohaga.

3.4 Paaris- ja paaritu funktsioon

Funktsiooni $y=f(x)$ nimetatakse paarisfunktsiooniks, kui iga x korral funktsiooni määramispiirkonnast kehtib võrdus $f(-x)=f(x)$. Funktsiooni $y=f(x)$ nimetatakse paarituks funktsiooniks, kui iga x korral selle funktsiooni määramispiirkonnast kehtib võrdus $f(-x)=-f(x)$. [4, lk 75, 76]

```
@interact
def paaris_ja_paaritu(fg=sin(x), x=(0,6,1)):
    px = point((x,fg(x)), color='red', size=30)
    negx = (-1)*x
    nx = point((negx,fg(negx)), color='purple', size=30)
    fgg = plot(fg, (x,-2*pi,2*pi))
    show(fgg+nx+px)
```



Illustratsioon 7: paaris- ja paaritu funktsioon

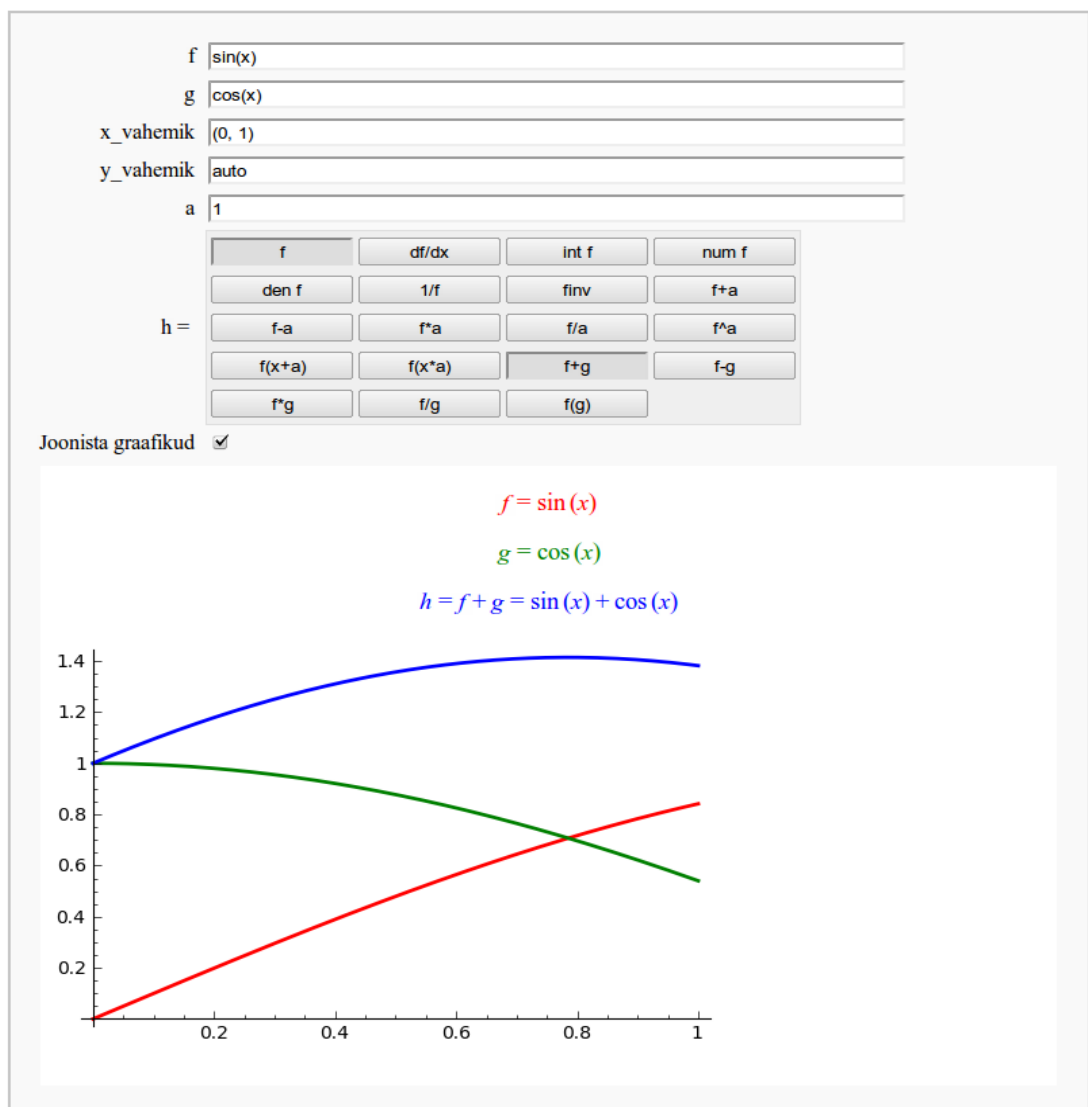
Jooniste interaktiivseks uurimiseks kasutatakse Sage'is käsku `@interact`, mille abil saab anda kasutajale võimaluse muuta näiteks funktsiooni ja tema muutuja väärtusi.

Illustratsioonil 7 kasutatakse funktsiooni muutmiseks lahtrivälja ja muutuja x väärtustamiseks liugurit. Muudetavad nupud on programmeeritud kasutaja poolt defineeritud funktsiooni `paaris_ja_paaritu` abil, kus algfunktsiooniks on seatud siinusfunktsioon ja teiseks vaadeldavad muutuja x väärtused. Saamaks graafikul aru, kas funktsioon on paaris või paaritu, loome käsu `point()` abil punkti px koordinaadid positiivse x ja negatiivse x puhul nx , mis aitavad joonise uurimisel leida, kas funktsiooni väärtus on iga muutja puhul sama või erinev. Käsk `point()` võtab esimeseks argumendiks punkti koordinaadid. Selguse mõttes värvime punktid erivärvi ja punkti suuruseks määrame 30. Kuna teame, et siinusgraafik on perioodiline perioodiga 2π , siis joonistuse käsule `plot()` seame x -väärtuste vahemikuks

-2π kuni 2π . Lõpuks liidame punktid ja funktsiooni graafiku kokku ja anname summa käsule `show()` argumendiks, mis paneb punktid ja graafiku ühele joonisele.

Nihutades liugurit, saab muutuja x vahemikus 0 kuni 6 täisarvulisi väärtusi, mille tagajärjel punktid p_x ja n_x muudavad oma asukohti. Nii graafikut uurides ning võttes arvesse funktsiooni paaris- ja paaritu funktsiooni definitsiooni, saame teada, kas funktsioon on paaris- või paaritu funktsioon. Samas kui siinusgraafikuga on kõik selge, saame ka muid funktsioone uurida, sisestades lahtrisse teise suvalise funktsiooni.

3.5 Funktsioonide sõltuvuse uurimine

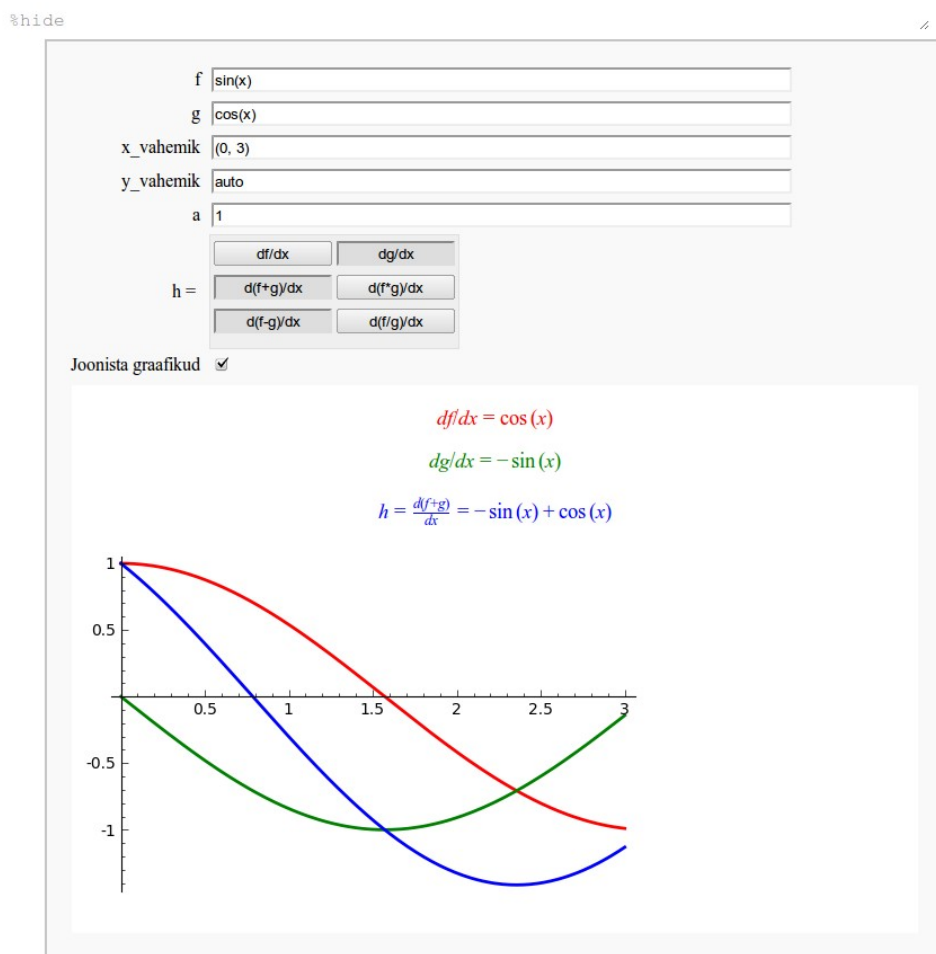


Illustratsioon 8: funktsiooni uurimine [12]

Illustratsioon 8 kujutab funktsiooni sõltuvuse uurimist kasutades Sage'is eeldefineeritud funktsiooni ja käsku `@interact`, mille abil kuvatakse muudetavad lahtrid ja nupud. Antud lahendusse saab sisestada kaks funktsiooni, graafiku kujutamiseks muutuja x -vahemikku ja funktsiooni väärtuste vahemikku. Algselt on määratud y -vahemikuks `auto`, mis tähendab, et y -telje kõrgus muutub automaatselt sõltuvalt funktsiooni „kõrgusest“. y -vahemikku saab ka ise määrata, andes vahemiku samal kujul nagu muutjale x . Defineeritud on ka arv a , mille algväärtus on 1, kuid seda saab ka muuta.

Edasi on defineeritud funktsiooniga f tehtavad tehted, nagu diferentseerimine, integraali leidmine, lugeja ja nimetaja leidmine, funktsioon astmes miinus üks, arvu liitmine, arvuga korrutamine, funktsiooni argumendile arvu liitmine jne. Samuti on uuritud kahe funktsiooni vastastikmõju, kus on uuritud funktsioonide liitmist, lahutamist, korrutamist, jagamist ja ühele funktsioonile teise argumendiks andmist.

Iga valiku puhul kuvatakse mõlema funktsiooni kirjutusviisid ja nendega sooritatud tehe. Viimasena kujutatakse kahe funktsiooni algkuju graafikud ja muudetud funktsiooni graafik. Graafikute joonistamist saab ka keelata, võttes linnukese Joonista graafikud käsu juurest.



Illustratsioon 9: funktsiooni tuletise uurimine

Funktsiooni tuletiste jaoks võib samamoodi luua uurimispupud ja graafikud. Mõlema ülesande puhul on Sage'is vaja kirjutada pikk kood, mis võib ülesannete lahendamisel muutuda segavaks asjaoluks – kogemata võib õpilane koodi lahtris midagi muuta ja ülesanne ei toimi. Eelneva olukorra vältimiseks tuleb sisestuslahtri esimesse ritta sisestada käsk `%hide`, mis peidab koodi, nähtavaks muutmiseks tuleb sellel uuesti klõpsata.

3.6 Arvu juurimine

Uurime edasi arvu ruutjuurt kõrgemate juurteni. Selleks võtame kasutusele järgmise tööriista, milleks on käsk `animate()`, mis kuvab ette antud elemendid viivitusega. Illustratsioonil 10 on kaks objekti, mida on vaja kuvada – valem ja graafik. Selleks on tarvis määrata kaks tühja graafiku muutujat `G` ja `H` ning loendi muutuja `v`. Animeerimiseks tuleb luua rohkem kui üks element, milleks kasutame `for` tsüklit, kus anname muutujale `n` väärtusi lõigus `[2, 12]`. Kombinatsiooniga `G +=` liidetakse tsükli käigus loodavad graafikud eeldefineeritud muutujasse `G`. Graafikute koostamisel antakse `x` väärtusteks lõik `[1, 3]` ning värvuse koodi loomisel kasutatakse tsüklimuutujat. Teiseks muutuvaks elemendiks on valemi kuvamistekst, kus käsule `text()` antakse valemi kuju, asukoha koordinaadid, fondi suurus ning samuti nagu graafikute puhul muutub ka valemite värv.

Kui vajalikud elemendid on loodud, liidetakse käsuga `append()` kokku graafiku muutujad eeldefineeritud muutujasse `v`. Edasi antakse käsule `animate()` argumentideks elementide loendi `v`, `y`-telje maksimaalne väärtus, mis omistatakse muutujale `animation`. Kogu vajaliku kuvamiseks kasutatakse käsku `show()`, mille argumentideks animatsiooni muutuja `animation` ja viivitus `delay`.

Antud ülesande illustratsioonil 10 näeb, kuidas juured lähenevad teatud kindlale piirile. Juurte asemel võime uurida ka muid funktsioone ühel graafikul konkreetses muutujate lõigus.

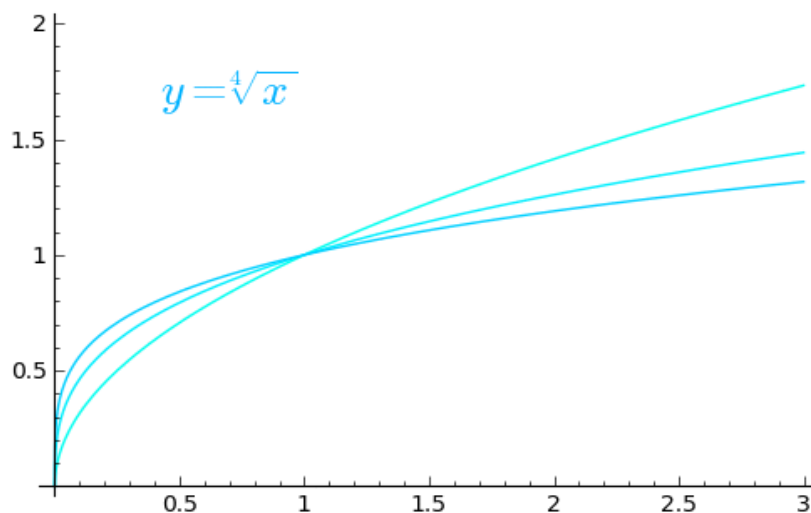
```

v = []
G = Graphics()
H = Graphics()

for n in range(2, 13):
    G += plot(x ^ (1 / n), [0, 3], hue=0.45 + 0.02 * n)
    H = text(
        "$y=\sqrt[%s]{x}$" % n, (0.7, 1.7),
        hue=0.47 + 0.02 * n, fontsize=20
    )
    v.append(G + H)

animation = animate(v, ymax=2)
show(animation, delay=40)

```



Illustratsioon 10: arvu juurimine

3.7 Integraali leidmine

Eelnevalt olen kasutanud juba palju tuletise leidmist, millega seostub integraali leidmine. Diferentseerimine (ehk tuletise leidmine) ja integreerimine on teineteise pöördoperatsioonid funktsioonide hulgas.

Avaldist $F(x) + C$, kus $F'(x) = f(x)$ ja C on mingi konstant, nimetatakse funktsiooni $y = f(x)$ algfunktsioonide üldavaldiseks ehk määramata integraaliks ja tähistatakse sümboliga $\int f(x) dx$. [7]

```

t = var('t')
v(t) = t^2 + 2*t + 2
vi(t) = integral(v,t)
vt(t) = diff(vi)
html("Algfunktsioon on <font color='red'>$v(t) = t^2+2t+2$</font>")
html("Integreerimise tulemus:")
html("<font color='blue'>$V(t) = \\int v(t)dt = %s$</font>" %latex(vi(t)))
html("Kontrolliks leiame tuletise integreerimise tulemusest: ")
html("<font color='green'>$\\sideset{}{'}V(t) = %s$</font>" %latex(vt(t)))

```

```

Algfunktsioon on  $v(t) = t^2 + 2t + 2$ 
Integreerimise tulemus:
 $V(t) = \int v(t)dt = \frac{1}{3}t^3 + t^2 + 2t$ 
Kontrolliks leiame tuletise integreerimise tulemusest:
 $V'(t) = t^2 + 2t + 2$ 

```

Illustratsioon 11: integreerimine

Illustratsioonil 11 võtame integreerimiseks funktsiooni $v(t)$. Sage'i käsk integreerimiseks on `integral()` või `integrate()`, millele on määrata integraali leidmiseks ette antud funktsioon ning integreerimismuutuja. Kontrollimaks diferentseerimise ja integreerimise pööratavust leiame integraali tulemusest tuletise käsuga `diff()`. Ülesande lahendamise tulemuse loetavamaks tegemiseks kasutan `html` ja LaTeX koodi ka koodilahtris.

Lõigus $[a, b]$ integreeruva funktsiooni $f(x)$ algfunktsiooni muutu, mis vastab argumenti muudule $\Delta x = b - a$, nimetatakse funktsiooni $f(x)$ määratud integraaliks rajades a -st b -ni ja tähistatakse sümboliga:

$$\int_a^b f(t)dt \quad [7]$$

3.8 Kõvertrapetsi pindala

Määratud integraal kujutab endast integraalsumma piirväärtust ja on arvuliselt võrdne sellele integraalile vastava kõvertrapetsi pindalaga. [5, lk 226]

Kõvertrapets on tasandiline kujund, mis on piiratud x -teljega, vertikaalsete sirgetega $x=a$ ja $x=b$ ning funktsiooni $y=f(x)$ graafikuga. [5, lk 224]

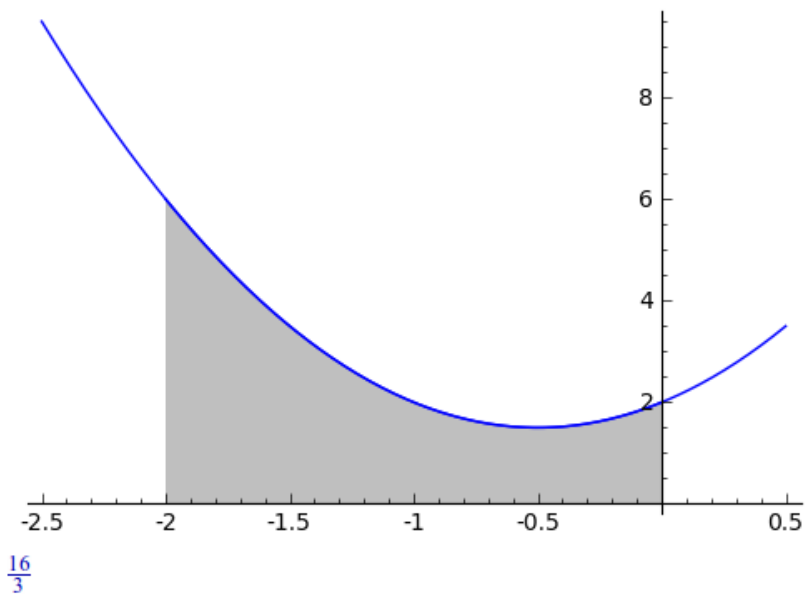
Eelnevat silmas pidades saame Sage'is luua ühe kõvertrapetsi graafiku ning leida selle pindala.

Illustratsiooni 12 kujutatud näite puhul eeldame, et funktsiooni väärtus iga x korral on positiivne, mistõttu on funktsiooniks valitud ruutfunktsioon $v(t)$.

```

t = var('t')
v(t) = 2*t^2 + 2*t + 2
a=-2; b=0
p=plot(v,a,b,fill=true)
pp=plot(v,-2.5,0.5)
vi(t) = integral(v,t,a,b)
(p+pp).show(); vi(t)

```



Illustratsioon 12: kõvertrapetsi pindala leidmine

Koordinaatteljestikul on kujutatud funktsioon $v(t)$, kus on liidetud sellest funktsioonist kaks graafikut ühele koordinaatteljestikule. Esimese graafiku koostamisel on argumentideks funktsioon, integreerimisrajad ning neljandaks kõvertrapetsi värvimine. Kui soovitakse eristada erinevaid tekkivaid kõvertrapetseid, siis saab anda käsule `plot()` lisaks argumenti `fillcolor`, mille väärtuseks on värvi nimetus. Teiseks graafikuks on funktsiooni laiem pilt.

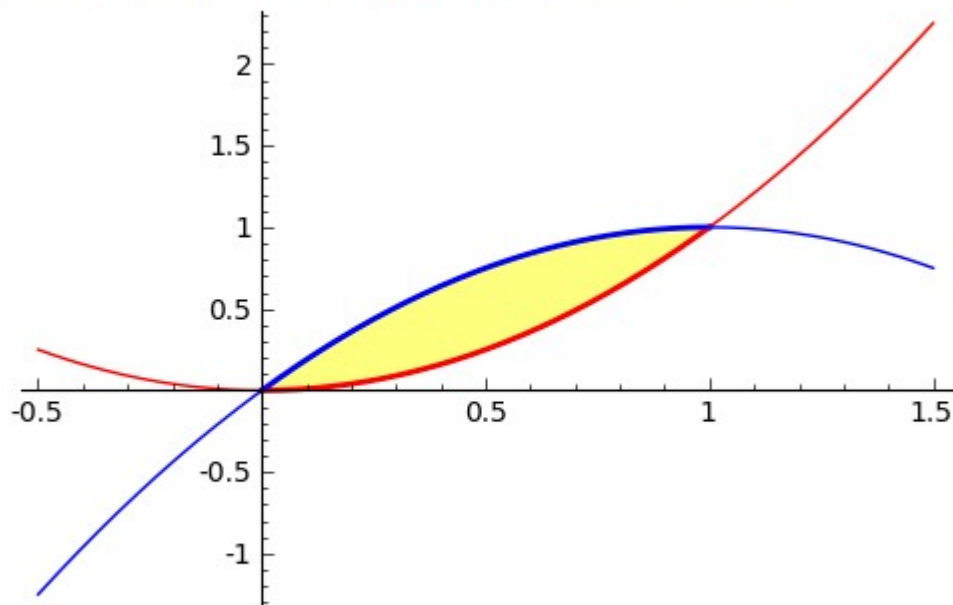
Leides nüüd määratud integraali, siis erinevuseks määramata integraali leidmisest on, et käsule `integral()` antakse argumentideks ka rajad a ja b .

Kui uuritavas piirkonnas on kõik funktsiooni väärtused negatiivsed, siis peaksime integraali tulemusest leidma käsuga `abs()` absoluutväärtuse.

3.9 Kahe joone vaheline pindala

Järgnevalt vaatame olukorda, kus pind tekib kahe funktsiooni graafiku vahele [39].

Pindala kahe joone $y=x^2$ ja $y=2x-x^2$ vahel



$$\text{Pindala} = \int_0^1 [f(x) - g(x)] dx = \frac{1}{3}$$

Illustratsioon 13: kahe funktsiooni vaheline pindala

Illustratsioonil 13 võime näha, et kui lahutame sinise joone aluse pindala punase joone alusest graafikust, siis saame fikseeritud lõigul $[0,1]$ kollase ala pindala kahe funktsiooni vahe integreerimisega. Et täidetaks ainult kahe joone vaheline piirkond, on vaja Sage'is graafiku koostamise käsule ette anda argumendiks `fill` ja selle väärtuseks funktsioon, milleni värvida. Antud juhul on `fill=x^2` ja `fillcolor='yellow'`.

Siin toodud näited on kättesaadavad ka töölehel <https://193.40.120.134:8050/home/pub/0>.

4. Sage'i server

Sage'i saab katsetada aadressil <http://demo.sagenb.org/>, mis on avalik Sage'i server. Sage'ist mingi ülevaate saamiseks võib seal käske proovida, kuid töölehti ei ole mõttekas seal luua, sest demoserveris kustutatakse mingil ajal liigsed töölehed. Seega on mitmed ülikoolid seadnud oma sisevõrku eraldi Sage'i servereid. Eraldi hallatavate serverite eeliseks on, et uuema versiooni tulekul saab vana asendada ajal, kui pole aktiivset õppetööd ja töölehtede kasutamist. Teiseks eeliseks on töölehtede varukoopiate loomine. [13]

Oma Sage'i serveri kasutamine väldib ülekoormuse teket, mis avaliku Sage'i serveri puhul võib aeg-ajalt tekkida, kuna paljude kasutajate mahukamate arvutuste ja 3D jooniste käivitamisel on vaja kasutada palju ressursi. Sõltuvalt Sage'i kasutatavate õpilaste arvust võiks Sage'i serveri sujuva töö tagada paar gigabaiti muutmälu, mis on põhiline riistvaraline nõue.

Sage'i saab alla laadida aadressilt <http://www.sagemath.org/download.html>, kus on palju peegelservereid. Valides ühe neist, avaneb Sage'i installeerimispakettide valik erinevatele platvormidele ja kahes erinevas olekus. Windowsi keskkonnas Sage ei tööta, kuid selle lahendab virtuaalmasin (näiteks [Vmware](#)), mis peab olema eelnevalt installeeritud. Seejärel tuleb laadida kokkupakitud fail `sage-vmware-x.y.zip`, lahti pakkida ning käivitada fail `sage_vmx` ja Sage alustab oma tööd. Virtuaalmasinas käivitatava Sage'i puhul peab tähele panema, et kasutatava arvuti muutmälu vajadus on suurem, kuna on vaja jookсутada Windowsi operatsiooni süsteemi ja Sage'i.

Ühe võimalusena Sage'i proovimiseks oma masinas on alla laadida buutiv Sage LIVE iso fail ning kirjutada plaadile või käivitada virtuaalmasinas. Iso failid sisaldavad kogu Sage'i töökeskkonda ja lisaks veel muid tööks vajalike programme, mis põhinevad väikesel Linux distributsioonil Lucid Puppy. [14]

Operatsiooni-süsteemidele Linux ja Apple Mac OS X on kõige parem alla laadida Sage'i binaarne distributsioon, mis sisaldab eelnevalt valmis ehitatud Sage'i versiooni. Kui aga laadida Sage'i lähtekoodi variant, siis selle ülesseadmine on pikemaajalisem ja keerulisem protsess. Pakkinud lahti Linuxis binaarse versiooni Sage'ist, saab seda tööle panna käsurealt `./sage`. Lahtipakitud Sage'i kausta saab tõsta suvalisse kohta ja jätkuvalt sealt käivitada.

Sage'i käivitamiseks suvalisest kaustast võiks lähtefailid viia fikseeritud kausta (näiteks `/opt`). Seejärel luua sümbolnimi järgnevalt:

```
ln -s /opt/sage-x.y.z /usr/local/bin/sage
```

Uute Sage'i versioonide saabumisel saab lihtsalt kustutada varasem sümbolnimi ning teha uue Sage'i versiooni kausta juurde. [15]

Andes Sage'i käsureal käsu `notebook()`, käivitatakse töölehe veebliides lokaalsena ehk siis käivitatavana kohalikust masinast, kasutades brauseris aadressitena `localhost:8000` või `127.0.0.1:8000`, mis on hea enda masinas testimiseks,

kuid mis ei too välja Sage'i kogu võimsust. Tegemaks Sage'i lokaalses kohtvõrgus kättesaadavaks ka teistele, peab käsule `notebook()` andma argumendiks lokaalse masina kohtvõrgu ip-aadressi ning pordi. Veebiliidesesse turvaliseks sisselogimiseks peab lisaargumendina andma `secure=True`, mis käsu esmakordsel käivitamisel genereerib etteantud ip-aadressi ja pordi jaoks SSL sertifikaadi.

```
sage: notebook(interfaces='', address='', port='', secure=True)
```

Jakobsoni koolis valisin Sage'i serveriks vanema arvuti, millel liigikaudu 100GB kõvaketas ja muutmälu 1Gb. Seadsin arvutile staatilise ip-aadressi 192.168.1.5 ning tulemüüris suunasin välise ip-aadressi 193.40.120.134 sisemisele koos pordi numbriga 8050.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli kirjeldada Sage'i kui avatud lähtekoodiga matemaatika töökeskkonda ja selle kasutatavust üldhariduskoolides. Võib öelda, et Sage'i on matemaatikas hea kasutada teooria kirjeldamisel, valemite koostamisel, arvutuste tegemisel ja graafikute joonistamisel. Töös on vaadeldud mitmeid matemaatika teemasid, mis leidsid kõik Sage'is lahenduse. Sage'i saab täiesti tasuta alla laadida, ning tööle seadmiseks on vaja leida vastavalt kasutajate hulgale mõni arvuti.

Sage võib tunduda alguses algtasemel arvutikasutajale keeruline ja aeganõudev, kuid selleks on Sage'il avalike näidetega server, kus on koostatud mitmeid matemaatika-ülesandeid. Samuti kasutades natuke Sage'i ja katsetades saab uute töölehtedega luua häid õppematerjale, mis on tehtud ühes kohas.

Kuna Sage'il on palju kasutajaid üle maailma, siis on juba loodud palju häid töölehtede näiteid. Sage'i arendajad kutsuvad üles kõiki, kes on loonud uusi ja paremaid töölehti või täiendanud olemasolevaid, laadima need uude avalikku Sage'i serverisse aadressil www.hssagemath.org, et avatud lähtekoodiga Sage'i kogukond võiks kasvada ning sellest suur abi oleks matemaatika õpetamisel. Sage on pidevas arengus ning uusi algoritme ja võimalusi lubatakse juurde teha pidevalt.

Summary

Using Sage for Teaching High School Mathematics

Bachelor's Thesis

Madis Pärna

By investigating how to find educational programs based on mathematics, we can find several programs e.g. Maple, Mathematica etc. which are by nature very good, but cost a large amount of money to purchase. The majority of schools don't have the necessary funds to purchase programs designed specifically for mathematics, therefore we have to look into freeware programs with an open source code.

There are several freeware mathematical programs available online, but unfortunately they are very specific and hardly ever meet all the requirements of solving the problems required. One that stands out from all the rest is Sage, which is an extremely powerful application that combines various freeware based software packages to create a unified user interface and is available to download for free with the option of setting it up within a LAN.

The purpose of the given dissertation is to describe Sage as an open source mathematical working environment and its suitability in general education. It is safe to say that Sage fits the use of creating graphs, mathematical expressions and describing the theory behind maths. The work was carried out to look at many mathematical themes, all of which were solved within Sage.

Sage might seem very complicated and time consuming to learn for a novice computer user. With this in mind Sage's developers have set up a public server, which contains several sample worksheets of mathematical problems. Likewise spending some time using Sage it is possible to create worksheets that make excellent study material and is all created centrally.

Since Sage has many users across the world, there are a lot of examples of great worksheets available. The developers of Sage encourage everybody who has created new or improved any existing contributions to upload them to Sage's new public server at www.hssagemath.org, so the open source Sage math community could grow and so it would provide help in teaching maths. Sage is in a stage of constant development to improve by allowing the creation of new computational algorithms and features.

The examples of code given in the thesis can be found at: <https://193.40.120.134:8050/home/pub/0>.

Kasutatud kirjandus

1. Sage'i veebileht, www.sagemath.org, [1.06.2011].
2. Sage'i server, demo.sagenb.org, [1.06.2011].
3. Sage'i server keskkooli teemadel, nb.hssagemath.org, [1.06.2011] ja hssagemath.org, [1.06.2011].
4. L. Lepmann, T. Lepmann, K. Velsker, Matemaatika 11. klassile, Tallinn, 2001.
5. T. Tõnso, A. Veelmaa, Matemaatika XII klassile, Tallinn, 2000.
6. L. Lepmann, T. Lepmann, H.-M. Varul, Ülesandeid gümnaasiumi matemaatika lõpueksamiks valmistumisel, Tallinn, 2006.
7. Integraali õppematerjal, <http://www.crjg.vil.ee/~marika/matemaatika/integraal/>, [1.06.2011].
8. Sage vikipeedias, http://en.wikipedia.org/wiki/Sage_%28mathematics_software%29, [1.06.2011].
9. Sage'i õppevideod, <http://showmedo.com/videotutorials/video?name=2450010&fromSeriesID=245>, [1.06.2011].
10. Latex käsud, <ftp://ftp.ams.org/ams/.../amsmath/short-math-guide.pdf>, [1.06.2011].
11. Sage'i soovitus, <http://www.sagemath.org/doc/reference/index.html>, [1.06.2011].
12. Sage avalikust serverist funktsiooni uurimise näide, <http://www.sagenb.org/home/pub/340/>, [1.06.2011].
13. Sage'i serveri informatsioon, <http://wiki.sagemath.org/SageServer>, [1.06.2011].
14. Sage'i serveri installeerimisjuhend, <http://wiki.sagemath.org/DownloadAndInstallationGuide>, [1.06.2011].
15. Sage'i binaarse versiooni allalaadimislink, <http://www.sagemath.org/doc/installation/binary.html>, [1.06.2011].
16. GAP, www.gap-system.org/, [1.06.2011].
17. Maxima, <http://maxima.sourceforge.net/>, [1.06.2011].
18. Singular, <http://www.singular.uni-kl.de/>, [1.06.2011].
19. MPIR, <http://www.mpir.org/>, [1.06.2011].

20. MPFR, <http://www.mpfr.org/>, [1.06.2011].
21. NTL, <http://www.shoup.net/ntl/>, [1.06.2011].
22. mpmath, <http://code.google.com/p/mpmath/>, [1.06.2011].
23. PARI/GP, <http://pari.math.u-bordeaux.fr/>, [1.06.2011].
24. mwrank, <http://www.warwick.ac.uk/~masgaj/mwrank/>, [1.06.2011].
25. SymPy, <http://code.google.com/p/sympy/>, [1.06.2011].
26. GiNaC, <http://www.ginac.de/>, [1.06.2011].
27. Symmetrica, <http://www.algorithm.uni-bayreuth.de/en/research/SYMMETRICA/>, [1.06.2011].
28. ATLAS, <http://math-atlas.sourceforge.net/>, [1.06.2011].
29. BLAS, http://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms, [1.06.2011].
30. LAPACK, <http://www.netlib.org/lapack/>, [1.06.2011].
31. NumPy, <http://numpy.scipy.org/>, [1.06.2011].
32. LinBox, <http://www.linalg.org/>, [1.06.2011].
33. IML, <http://www.cs.uwaterloo.ca/~astorjoh/iml.html>, [1.06.2011].
34. GSL, <http://www.gnu.org/software/gsl/>, [1.06.2011].
35. NetworkX, <http://networkx.lanl.gov/>, [1.06.2011].
36. SciPy, <http://www.scipy.org/>, [1.06.2011].
37. FLINT, <http://www.flintlib.org/>, [1.06.2011].
38. R, <http://www.r-project.org/>, [1.06.2011].
39. Funktsioonide vaheline viirutus, <http://nb.hssagemath.org/home/pub/21/>, [1.06.2011].